Using Alignments in Automatic Paraphrase Production to Combat Data Sparsity in Question Interpretation for a Virtual Patient Dialogue System

Undergraduate Research Thesis

Presented in Partial Fulfillment of the Requirements for graduation "with Honors Research Distinction in Linguistics" in the undergraduate colleges of The Ohio State University

by
Sarah Ewing

The Ohio State University
May 2019

Project Advisor: Professor Michael White, Department of Linguistics

1. INTRODUCTION

In this paper we introduce a new method for leveraging alignments to generate question paraphrases automatically. Successful question paraphrasing has positive implications in a downstream, natural language question answering task. However, the applications for automatic paraphrase generation go beyond our downstream task. Producing paraphrases can benefit natural language tasks such as summarization, evaluation of machine translation, and beyond.

We work within a dataset from an existing virtual patient dialogue system (Danforth et al., 2009, 2013). The data consists of turns from Ohio State University medical students interacting with the virtual patient, Jim Wilkins, who suffers from back pain. The task performed in each conversation with Mr.Wilkins is that of a medical interview about the patient's current health concerns and medical history. The goal of the system as a whole is to answer questions correctly and fluently so as to not disrupt to flow of conversation.

We will first explain in detail the virtual patient dialogue system and its flaws. Then we will discuss previous attempts to automatically generate paraphrases both within and outside of the project. We will discuss the benefits of using alignments in paraphrase production, and introduce a new process for generating paraphrases. We find success in creating a number of paraphrases and analyze the potential implications of this on the success of the virtual patient project as a whole. We also suggest methods for increasing the success of our system.

This project was aided significantly by the efforts of the entire virtual patient dialogue system team. In particular, Amad Hussein produces automatic alignments from ELMo representations used in paraphrase production here. David King compiles downstream assessment tasks for paraphrases produced, as well as producing alignment dictionaries, which are tools used to increase paraphrase production.

2. BACKGROUND

2.1 The Virtual Patient Project

Standardized Patients (SPs) are human actors who play patients in the setting of a medical interview for the purposes of training and assessing medical students. SPs have a controlled patholo-

gy and medical history, but their performance between medical students can be inconsistent and it can be expensive to hire human actors to test students. Additionally, inconsistent testing conditions can make evaluating medical students and comparing their performance more difficult. To attempt to fix these problems the Department of Family Medicine at OSU developed a virtual patient dialogue system, which is currently used for medical students' practice only. The original system works by receiving typed, interview-style questions from medical students and responding appropriately. The latest version of the system uses spoken language input, but the data from this version of the system was not available for our work. The virtual patient performs the task of question answering by maintaining a list of possible question labels. Question labels are defined as the simplest form of each question the system is intended to answer. Each question label is paired with a canned answer about the topic. The system marks each novel input as a token of one of its known labels and returns the response it has associated with that label. There is a catch all, 'unknown' question label which prompts the virtual patient to respond by saying it didn't understand the question. When students ask extraneous questions, the virtual patient responds correctly with the 'unknown' label response. Therefore, every input question should have an appropriate response from the system.

The initial system used only a hand-written, pattern matching technique called ChatScript (Wilcox, 2011), which has relatively high rates of success in correctly matching a student's question to an appropriate label. However, ChatScript is relatively difficult to maintain as the needs of the system grow, and as new virtual patients are developed. Jin et al. (2017) implemented instead an ensemble of word- and character- based convolutional neural networks (CNNs) to do the job of question answering. To determine what label a novel question belongs to, the system performs multi-class classification. Therefore, a novel question is classified by the system's decision of its most probable label. The system also gives a confidence score for its decisions. This system attained 79% accuracy on the question answering task, which was comparable to the original ChatScript system. However, the error profiles were much different between the two systems, and the group found that combining the systems by using their confidence scores to decide between their label choices improved overall accuracy nearly 10% with a 47% reduction in error over the ChatScript system alone.

2.2 Paraphrase Generation

This improvement was significant, but still left something to be desired in the error reduction of the CNN ensemble. The training data came from 94 dialogues of medical students practicing with the virtual patient dialogue system for a total of 4330 user turns. There were 359 unique question labels with a mean of 12 instances per label, median of 4, and large standard deviation 20. Only the 8 most frequent labels account for nearly 20% of the data, whereas the 265 least frequent labels also account for around 20% of the data. This means the label frequency has an extremely long tail as shown in Figure 1. Importantly, many of the errors in the CNNs performance occurred on these rare labels. Its accuracy on the least frequent quintile was only 46.5%. ChatScript performs at 70% accuracy on rare labels which is almost 10% below its average. To combat this issue of uneven data sparsity we attempt to introduce automatic paraphrase generation, especially for rare labels, to augment the training data.

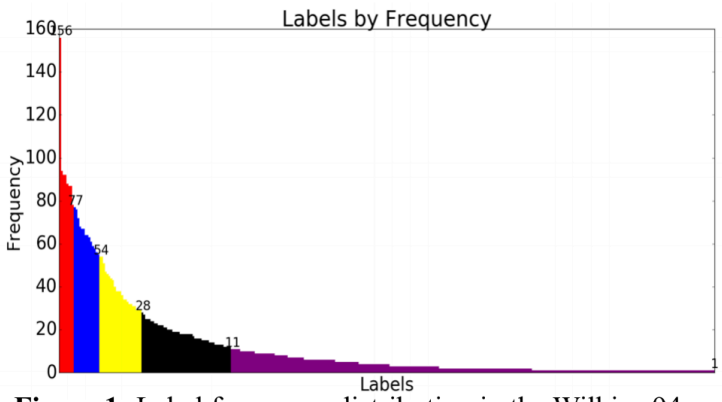


Figure 1: Label frequency distribution in the Wilkins 94 Dialogues database. Labels are distinguished by quintile. The 265 labels of the least frequent quintile are shown in purple. All have 11 or fewer tokens of that label. Reproduced from Jin et al. (2017).

The first attempt in that direction is made by Jin et al. (2018) who implement memory-augmentation in the CNN stack and who use neural machine translation (NMT) to generate paraphrases. The configuration of the memory-augmented CNN (MA-CNN) is described in their paper, and was motivated by its ability to do one-shot learning, which is inherently designed for

smaller training sets. However, the MA-CNN is still biased towards similar-sized training groups for classification categories. Thus our long-tailed dataset still presents problems, and paraphrase generation will hopefully improve even the MA-CNN system.

The methods used to create paraphrases were two-fold. First, lexical swaps were made where the new word created a candidate paraphrase with meaning sufficiently similar to the original question's. Jin et al. use three sources of possible swaps and evaluate each source by how frequently their suggested swaps occur across paraphrases in the Wilkins training data. They then produce paraphrases on only the rare data and using only the most successful swaps across sources.

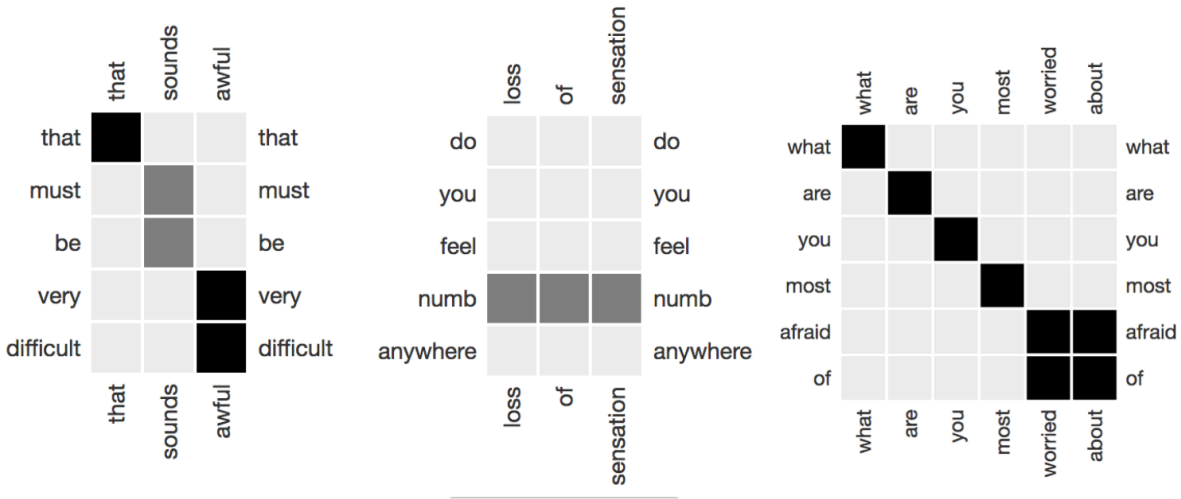
Second, candidate paraphrases were formed through NMT. Mallison et al. (2017) introduce a method of paraphrasing using NMT with a neural encoder-decoder framework for machine translation they name PARANEX. Their system encodes a source sentence into a vector-representation and then uses a sequence-to-sequence model to transform the representation into a sentence of a pivot language. The pivot sentence is then translated back into the original language through the same process creating a candidate paraphrase to the source. The motivation behind this process is that an advanced method of translation should preserve the overall meaning and form of a sentence. However, it is also likely to create a different sentence from the original because translation is inherently not one-to-one mapping. Mallison et al. find the best results by choosing the k-best translations in up to three distinct pivot languages.

Jin et al. chose a single pivot language, German, and take the 10 best pivot sentences. These ten German translations were then translated back into English, again selecting the top ten translations for each. The resulting 100 candidate paraphrases were reduced to remove duplicates and ranked based on the initial accuracy judgement of each direction of translation and based on frequency before removal of duplicates. The group makes significant efforts, automatic and manual, to filter paraphrases for both accuracy and novelty. They find that both augmenting the training data with their paraphrases and using a MA-CNN instead of the original CNN ensemble improved the systems accuracy on rare labels by almost 10%. However, this combination resulted in lower accuracy overall. Therefore, we implement new methods which use alignments to generate candidate paraphrases in an attempt to continue augmenting the training data and im-

proving the downstream accuracy of the current virtual patient dialogue system. We believe use of alignments will produce paraphrases which reflect the domain-specific language use patterns, because they specify our knowledge of interchangeable strings across paraphrases.

2.3 Alignments

Gokcen et al. (2016) introduce a corpus of manually annotated gold standard word alignments referred to as gold alignments. They extract pairs of sentences from the virtual patient dialogue system's data and perform alignment annotation. Each pair is judged to be or not be a set of paraphrases. They make alignments that are one word to one word (1-to-1), one word to more than one word (1-to-many), more than one word to one word (many-to-1), or more than one word to more than one word (many-to-many). For example, in Figure 2, *that* is aligned to *that* in a 1-to-1 alignment, and *awful* is in a 1-to-many alignment with *very difficult*. In Figure 3, *loss of sensation* is in a many-to-1 alignment with *numb*, and in Figure 4 *worried about* and *afraid of* are in a many-to-many alignment. These alignments can be marked as either 'sure' or 'possible'. The extensive annotation guidelines which dictate alignment decisions can be found in their paper (Gokcen et al. 2016). In total they annotate 942 sentence pairs of which 441 are



Figures 2,3,4: Alignments from the gold alignment corpus. *that sounds awful* and *that must be very difficult* are paraphrases as well as *loss of sensation* and *do you feel numb anywhere* and *what are you most worried about* and *what are you most afraid of*. The shaded squares in the grids represent aligned words. Black squares are sure alignments, and grey are possible alignments. Reproduced from Gokcen et al. (2016).

paraphrases. In comparison, if we run automatic alignments on our corpus where paraphrase pairs are considered to be any two questions of the same gold label, we would be aligning 191,070 paraphrase pairs. However, a gold standard set of alignments is extremely helpful for initiating our process.

Amad Hussein pursued automatic alignments to augment alignment numbers and to increase the applicability of alignments to new patients and other developments in the system. He uses ELMo embeddings, as introduced by Peters et al. (2018) to make automatic alignments. Word embeddings are vector representations of words filled with values determined by information from language models. We chose to use these embeddings because they are the state-of-the-art contextualized word embeddings at this time. Contextualized embeddings are important when producing paraphrases because they better at account for the meaning of words in each context in which it is found. Hussein determines paraphrases as any two questions in the data that have the same gold label; gold labels are annotated by humans as what the ideal label for a question was. The ELMo representation is found for each word in each question, and then the most similar representations are aligned across the pair in a greedy fashion. By the nature of ELMo representations, these alignments were only 1-to-1. Until improvements are made to allow non 1-to-1 alignments in this process, the automatic alignments are at a disadvantage for finding the correct alignment of paraphrases. Therefore we are less confident in the ELMo alignments, although we believe they have potential to aid future work.

3. RELATED WORK

3.1 Using Alignments for Paraphrasing

Fader et al. use automatic question paraphrasing as a step to improve their open domain question answering task and allow for the answering of completely novel questions with encouraging results. They begin by manually creating 16 'seed' question templates. Through automatic alignments in their dataset, they extract contiguous 'entity patterns' and 'relation patterns' aligned to specific variables in the question templates. The method they use for making automatic alignments is the MGIZA++ (Och and Ney 2003) implementation of IBM Model 4. IBM Model 4 makes alignments in order of input and takes account of word class, roughly equivalent to part of

speech, when inducing alignments. Fader et al. run this model in both directions and combine the results of each direction heuristically.

They also generate further question patterns from the structures of the questions in the data which align to the seed templates. Their paraphrase corpus comes from WikiAnswers where users have the opportunity to mark new questions as alternate-wordings of existing questions. Of the pairs marked as in this way, Fader et al. analyze a subset of 100 and find 55% to be valid paraphrases. They use alignments on all pairs to extract tuples which consist of relations, entities, and desired information. For example, in Figure 5, the seed question template *what is the r of e* has associated tuple *r(?, e)* and when the system encounters a question such as *What is the population of New York?*, the information from which fills the *r* and *e* variables in the templates will populate the tuple as follows *population(?, new-york)*. An answer to this question will fill the *?* variable. Additionally, paraphrases of *What is the population of New York?* such as *How big is NYC?* create new question templates by recognizing the same entity *new-york* and relation *population* in both questions. Thus the new template created has the form *how r is e*, and the entity *new-york* has two values *New York* and *NYC* as does the relation *population*, which has values *population* and *big*. The *?* variable can then be filled in all tuples that are paraphrases of a question with a known answer. Question answering then becomes a task of aligning a novel question to its paraphrase group and extracting the response from the group's tuple.

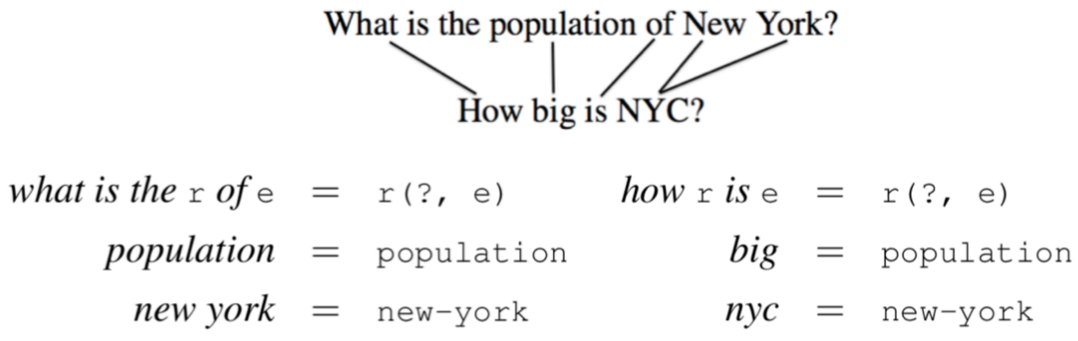


Figure 5: In this figure the alignment between *What is the population of New York?* and *How big is NYC?* produces a new question template, *how r is e* as well as the values *population and big* for the entity *population* and values *new york* and *nyc* for entity *new-york*. Reproduced from Fader et al. (2013).

4. ANALYSIS OF RELATED WORK

4.1 Applying Fader et al. to our Domain

We sought to test the applicability of Fader et al.'s question templates, used instead for automatic generation of paraphrase candidates, on our domain. We believe this method will produce better and more differentiated paraphrases than lexical swaps because the informative strings (entities and relations) will be preserved in the process of paraphrasing as opposed to trying to find semantically identical substitutes. As stated in their paper, their work makes significant improvements over other systems, "with more limited semantics, …at a very large scale in an open-domain manner" (Fader et al. 2013). Our task was to see if their success would generalize to our smaller domain of patient interviews.

The Fader et al. database consists of groups of question templates considered to be paraphrases. Matching each member of the group with each other member once created 5,137,588 question template pairs, made up of a left and right side. We find 109,360 of the left sides of these pairs to be unique. Of those, we find 1,637 to contain only words in the vocabulary of our virtual patient dataset. Note here that this qualification was intended to be extremely unrestrictive; we assumed question templates should theoretically contain only very common words to English because the more informative, and therefore less common, entities and relations have been extracted. The same common words should be found in our dataset. However, we found that only 1.5 percent of the templates pass this vocabulary filter. For example, templates with left hand sides shown in A) pass the vocabulary filter, but the templates with left-hand sides in B) do not.

A) $y all the time?
   $y abdomen?
   $y blood pressure?

B) $y alternator belt?
   $y 2005 malibu?
   $y basketball?

These 1,637 unique left sides which passed the vocabulary filter were found in 416,644 question template pairs. When applied to questions in our virtual patient data, 162,393 candidate paraphrases were formed. Because question paraphrases already existed in our virtual patient data as a result of multiple students asking questions of the same label, our first measure of accuracy for the candidate paraphrases was to check how many matched exactly to a question which already appeared in our data. Surprisingly, we found that number to be zero. To further evaluate these candidates, 200 were annotated for accuracy manually with a precision of less than 5%. In Table 1, the second row is an example of a good paraphrase created in by this method, and the first is an example of a poor candidate paraphrase.

Template	Match	Candidate Paraphrase	Accuracy Judgement
do $y family —> $y live at	do you have any close family	you have any close live at	Bad
do $y alone—> who do $y with	do you live alone	who do you live with	Good
do $y have —> are $y English	do you have any pain	are you English any pain	Bad
do $y like —> do $y have a baby	do you like your job	do you have a baby your job	Bad

Table 1: This chart shows paraphrases produced using the Fader et al. templates. The template pairs in the first column are separated by an arrow to represent the question matched by the side of the template before the arrow and the candidate paraphrases being produced by the side of the template after the arrow. Elsewhere, templates sides before and after the arrows are referred to as left and right hand sides of the templates respectively.

4.2 Analysis of Results

Our conclusion was that our dataset was too small, specific, and semantically rich for the methods used in Fader et al. Many of the errors found were likely due to the nature of the WikiAnswers corpus from which the alignments were extracted. Template pairs frequently inserted odd content into candidate paraphrases, as can be seen in rows three and four in Table 1. In our domain, nationality is of little importance, and here leads to a poor paraphrase. Additionally, having a baby was irrelevant to the question about our patient's happiness in his job. This problem

supports the suggested importance of question templates being predominantly free of informative strings such as *English* and *have a baby*.

Further, it seems the templates selected for short questions. The Fader et al. templates have an average of 2 words and one variable. The alignments these templates were created on aligned entities on average 2 words in length, and relations on average 1 word in length. Therefore, we would expect to see predominantly 4 - 8 word questions in their data. The average length of questions in the virtual patient dataset is just over 8 words. Thus, these templates seem to be an ill fit for our data. In their own analysis, Fader et al. find only 55% of the matched sentences which created template pairs to be true paraphrases. This causes significant noise in our data, but we would still hope to see a higher fraction of the true paraphrase templates leading to good paraphrases.

5. METHODOLOGY

5.1 Inducing Question Templates

We borrow the concepts of question templates and entity patterns from Fader's work but approach the problem differently. Although our dataset is much smaller than the WikiAnswers Corpus, we have the advantage of the corpus of human aligned paraphrases from our data referred to as gold alignments from the Gocken paper. We did not begin our paraphrase generation with seed templates. Instead, all templates were generated automatically. From gold aligned paraphrases, templates were created by removing 'informative', aligned, contiguous strings from each sentence and replacing them with variables. 'Informative' here is used to mean words which are not a member of a list of stop words, which are extremely common words of English, especially in questions. If either side of an alignment contained a word that was informative, both sides were replaced by a variable. Each variable was marked to match its aligned variable in the other half of the question template. For example, the alignments shown in Figure 6 lead to the template pair C:

C) ['is', '$0', 'worse', '$1', '$2', '$3'], ['is', '$0', 'just', '$1', '$2', '$3']

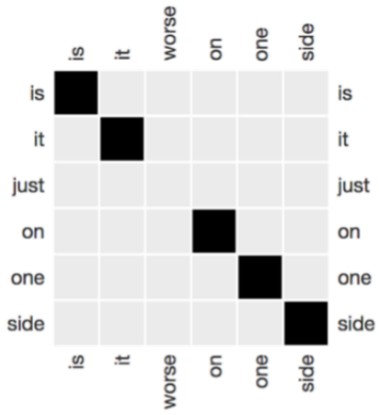


Figure 6: Image from gold alignments annotation interface. Here, *is it worse on one side* and *is it just on one side* are paraphrases. The filled boxes in the grid represent aligned content. For example *is* in the first sentence is aligned with *is* in the second sentence and no other words.

All alignments are replaced by variables except *is* aligned to *is*. This is because *is* is considered to be a stop word. *It* is not considered to be a stop word because the variable which replaces it frequently matches on non-pronominal phrases which create good paraphrase, such as *the pain* in the continued example. Once these question templates are generated, the variables match to any length string of words in questions from our training data. For example, C) matches on *Is the pain worse in the morning or at night* and produces the paraphrase *Is the pain just in the morning or at night*.

There were varying restrictions on what could match to each variable. Initially, variables were allowed to match on any continuous string. Later, 1-to-1 variables were distinguished from other variables in template generation. The 1-to-1 variables were marked with the part of speech (POS) tag of the word which they replaced in the initial aligned sentences. When matching to sentences in the training data, multiword variables (not 1-to-1) can still match to any continuous strings, but single word variables (1-to-1) must match to single words in the question with the correct corresponding POS tag. Further restrictions were explored on multiword variables which included restricting length and POS tags of acceptable matches. We explore these varying restrictions to measure the changes in precision and recall that they produce. We expect the precision, the percent of paraphrases which are good out of those produced, to increase as we enforce stricter restrictions on variables. We expect the recall, the percent of total good paraphrases we produce out of those possible, to decrease as we enforce restrictions. Although we do not have an exact number for the total number of possible paraphrases we could produce, we know that re-

ducing the number of good paraphrases we produce will reduce recall. We hope that the precision increases significantly but the recall decreases only minimally with each change.

Once a template matches to a question in the training data, the strings corresponding to variables in the left half of the template are extracted and inserted in the appropriate gaps of the right half. For variables with more restrictive features, these extracted strings may be altered before adding them into the paraphrase.

5.2 Altering Variables

To improve the quality of our candidate paraphrases we attempt to make changes to variables where deemed necessary. For example, from the aligned sentences in D) we extract the template pair E).

D) **Any trouble urinating?**
How's your urination?

E) ['Any', 'trouble', '$0/VBG'], ['How', '\'s', 'your', $0/NN].

F) **How's your urinating*
**Any trouble urination*

In E) we see that our $0 variable has a different POS tag in the left side of the template pair than in the right side, and this accounts for important information in the data. Although *urinating* and *urination* should clearly be aligned, the sentences in F) are bad sentences of English and we consider the implications of generating them. The questions, although ungrammatical, are understandable by humans. It is possible that medical students could ask ungrammatical questions, especially considering that English is not the first language for all students using this tool. In these ways, the sentences in F) may be useful to generate. However, in other downstream tasks such as an authoring tool or a natural language generator, these paraphrases would be harmful. Additionally, restricting the production of these paraphrases upholds a higher standard on paraphrasing more similar to human paraphrasing in L1 English speakers. In either case, we

want to produce the correct English paraphrases whether we leave in the questionable ones or not. When multiword variables are given a length restriction, there may additionally be an offset in length between a left and right side of a variable as well. In these situations we make efforts to alter a matched string to fit its right-side variable requirements.

To alter the POS tag of a word we attempt to lemmatize and then reform the words of a variable. We compile a reference 'dictionary' which acts as a sort of lookup table. Known lemmas can be considered the rows, and the POS tags of words found to have that lemma are the columns associated with a lemma's row. To transform a word to a different POS tag we simply lemmatize it and then search our table for the square which corresponds to its lemma and desired POS tag. An ideal example entry from the dictionary is given in G). Once we know our lemma is 'run', we can search the list of columns of POS tags to find the word associated to that square.

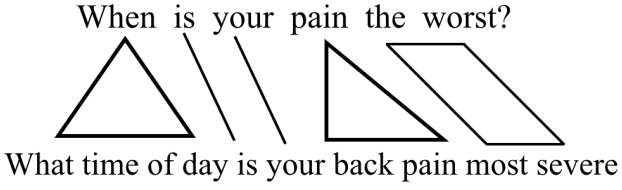
G)

	NN	**NNS**	**VB**	**VBZ**	**VBG**	**…**
…						
run	run	runs	run	runs	running	…
…						

If a question like *Any trouble sleeping?* matches the template in E) above, 'sleeping' should be lemmatized to 'sleep' and then the dictionary is searched for the POS tag 'NN' on lemma 'sleep' and should return 'sleep', giving us the paraphrase *How's your sleep?*.

For templates in which we implement restrictions on length, it is also possible to have a mismatch in the length of a variable from left to right side of a template. In these cases, the difference is less likely to cause a problem in the candidate paraphrase. For example, the template in I) comes from the aligned sentences in H).

H)



I) ['$0/[1]', 'is', 'your', '$1/[1]', '$2/[2]'], ['$0/[4]', 'is', 'your', '$1/[2]', '$2/[2]']

When matching on the original sentence, *When is your pain the worst?*, transferring the variables, as they are, into the right side of the template makes a good sentence of English, *When is your pain the worst?*. However, we can see that this is actually an exact repeat of the original sentence. To make the paraphrase differentiated from the original sentence, we want to obey the restrictions set by the variables.

We implement 'lookup' dictionaries like the one we use for POS changes of both gold and ELMo alignments. Instead of searching for new words on lemmas and POS tags, we search for new phrases by phrase and length desired for the new phrase. These dictionaries can be additionally implemented on each variable to create additional paraphrases replacing the variable with each distinct aligned string in the dictionary. For example, the lookup table shown in J) can be used on the substring *chronic health problems* of the question L). The substring matches to variable '$1' in the template K). Replacing the variable with each associated string in the dictionary entry in the right hand side of the template creates the list of candidate paraphrases given in M). This results in a significant growth in the number of candidate paraphrases produced. The alignments in these dictionaries are made from the Wilkins 94 dialogues dataset, and only between sentences which have the same human annotated gold label.

J)

...					
chronic health problems	medical conditions	health problems	medical problems chronic medical conditions	medical issues	...
...					

K) ['do', '$0', 'have', '$1'], ['what', '$1', 'do', '$0', 'have']

L) Do you have chronic health problems?

M) What chronic health problems do you have?

What medical conditions do you have?

What health problems do you have?

What medical problems do you have? …

6. DATA

6.1 Datasets

The database from which templates are formed is described in Gokcen et al. (2017). The data comes from a preliminary use of the virtual patient dialogue system. It consists of 104 dialogues between the system and medical students. There are 5437 individual turns in this data, and an average of 7 words per turn. The 289 unique labels in the data have an average of 17 turns. The gold alignments were done on 942 sentence pairs selected from sentence pairs which were the closest to the decision line of a binary classifier. The classifier was trained to distinguish paraphrases from non-paraphrases. Of the 942 pairs, 441 were determined to be paraphrases by the annotators.

The database on which matches were made, referred to as the Wilkins 94 dialogues dataset, was slightly different. This database consists of only 94 dialogues. Of the 4330 turns, 3288 are unique. The average number of words per turn is 8. There are 335 unique labels used, and each label has an average of 13 turns.

The difference in these databases is significant because of its impact on the candidate paraphrases. If we matched templates on the same data that they were created from, we would expect to make at least as many candidate paraphrases as we have templates. We should also expect to see at least that many candidate paraphrases which are repeats of existing questions. Instead, we have no lower limit on these values. Additionally, using a varied dataset for matching supports the ideal of applicability of the templates outside one dataset. Although we want these labels to be highly useful in our specific domain, we also want them to extrapolate to other virtual patients or related dialogue systems.

6.2 Methodology Stats

We make 117 templates, 113 of which are unique, when enforcing no restrictions on variables. Enforcing POS tag restrictions on only single word variables produces 250 templates, with 246 being unique. The more restrictive nature of these templates produces more templates for good reason. Templates that have identical left and right sides are thrown out, as they only

reproduce the questions already found in our data. When we add POS tagging, some templates which had identical sides with no restrictions become differentiated. When single word variables are marked for POS and multiword variables are marked for length and POS, 455 templates are produced, of which 446 are unique, again the more restrictive methods produce more templates. These numbers don't change when POS tag restrictions are dropped for multiword variables because we use the same templates for that process but don't require matching on POS tags.

The dictionary we use to alter the POS tag for variables has 5,600,837 lemmas in it. The dictionary of gold alignments we use to do variable swapping has 772 aligned phrases. 406 of these are multiword phrases and 366 are single words. Phrases in the dictionary are aligned to an average of 1.6 other phrases and single words are aligned to an average of 5 other words. Therefore, entries in the dictionary are aligned to an average of 3.2 other entries.

6.3 ELMo

If we implement automatic alignments we have the potential for much higher numbers of templates and candidate paraphrases. As stated, there are 191,070 paraphrase pairs on which to make automatic alignments. Using ELMo alignments on these pairs produces 63,963 template pairs, of which 63,932 are unique. Unfortunately, these templates only have single word alignments, but this shows the impact multiword automatic alignments could make on the paraphrase generation process.

Additionally, ELMo alignments produce an alignment dictionary like the gold alignment dictionary. The ELMo alignment dictionary has 1,166 entries and, together with templates only restricting POS tags for single word variables, produces 1,289,335 candidate paraphrases. This again would benefit from developing multiword variable swaps, although it is already producing a lot.

6.4 Paraphrases

As shown in Table 2, Implementing no restrictions on the template variables, called None in Table 2, produced 42,312 candidate paraphrases, of which 22,942 or 54 percent are unique. 100 of these paraphrases were found in the Wilkins 94 dialogues dataset. Additionally, we singly-annotate 114 unique candidate paraphrases and find 27 to be good, meaning this subset has 24

Restrictions	Templates	Unique Templates	Paraphrases	Unique Paraphrases	% Unique Paraphrases	Paraphrases in Wilkins Data	% Unique Good Paraphrases	Rare Unique Paraphrases	% Rare Unique Paraphrases	Expected Good Rare Paraphrases
None	117	113	42,312	22,942	54	100	24	6,083	26.5	1460
1-to-1 POS	250	246	1889	906	48	8	35	225	24.8	79
All POS	455	446	1962	257	13	36	41	76	29.6	31
1-to-1 POS, Other Length	455	446	3697	335	9	36	38	116	34.6	44
1-to-t POS, Variable Swaps	250	246	> 18,000,000	30,059	0.1	6	2	7708	25.6	154

Table 2: Results on paraphrase production using 5 different methods of restriction on variables. Methods are; None, templates with no restrictions. 1-to-1 POS, templates with POS tag restrictions on single word variables. All POS, templates with length and POS tag restrictions on all variables. 1-to-1 POS, Other Length, templates with POS tag restrictions on single word variables and only length restrictions on multiword variables. 1-to-1 POS, Variable Swaps, templates with POS tag restrictions on single word variables, and variable swapping on multiword variables. Each method is evaluated for the number of templates it produces, number of unique templates, number of candidate paraphrases it produces, number of unique candidate paraphrases, the percent of candidate paraphrases which are unique, the number of candidate paraphrases which are found in the training data, the percent of unique candidate paraphrases which are good based on annotation of a subset, the number of unique candidate paraphrases produced on rare labels, the percent of total candidate paraphrases produced which re on rare labels, and the number of candidate paraphrases on rare labels that are expected to be good based on the percent of total unique paraphrases which are good.

percent precision. Extrapolating, this rate of success would produce 5,406 novel paraphrases for the training data. Of the 22,942 unique candidate paraphrases made, 6083 of them were on rare labels. With consistent precision we would hope to see 1460 valid paraphrases on rare labels.

An example of a good paraphrase from this method is given in Figure 7. An example of a poor paraphrase from this method is given in Figure 8. In the poor candidate paraphrase the problem arises form the fact that both 'no' and 'any' are used as determiners and it is not proper English to use two consecutive determiners. Therefore we make changes to the templates to produce good paraphrases more frequently. In this case, our POS restrictions should restrict the variable '$0' from being a determiner because it directly follows 'No'.

When POS tag restrictions were implemented on single word variables 1,889 candidate paraphrases are produced. Of these paraphrases, 906 are unique which is 48 percent of the total

Original Aligned Sentences
What activities do you do?
What kind of activities do you do?

Template Created
['what', '$0', '$1', 'you', '$2'], ['what', 'kind', 'of', '$0', '$1', 'you', '$2']

Matched Sentence In Training Data
What treatments have you used so far?

Candidate Paraphrase Produced
What kind of treatments have you used so far?

Figure 7: An example of a successful process of paraphrase production using a method of no restrictions on variables.

Original Aligned Sentences
Do you have high blood pressure?
No diabetes? No high blood pressure?

Template Created
['do', 'you', 'have', '$0', '$1', '$2'], ['no', 'diabetes', '?', 'no', '$0', '$1', '$2']

Matched Sentence In Training Data
Do you have any ongoing medical problems?

Candidate Paraphrase Produced
*No diabetes? No any ongoing medical problems

Figure 8: An example of an unsuccessful process of paraphrase production using a method of no restrictions on variables.

number. Of these candidates, 8 were found already in the Wilkins data. We annotate 100 and find 35 to be good paraphrases. This would produce 309 new paraphrases for our training data. With 225 candidate paraphrases produced on rare labels we would hope to see 79 good paraphrases on rare labels.

As expected, we see from Figure 9 that implementing POS tag restrictions on single word variables prevented the production of a poor candidate paraphrase, increasing overall precision. However, the restriction also prevented the production of a good paraphrase as shown in Figure 10. The restriction of this good match is due to its strict enforcement of a noun following the determiner although an adjective following a determiner is equally acceptable in English. Additionally the POS tag restrictions require each variable to match only one word. These changes recall of our system in general by enforcing a higher standard of similarity between original aligned

Original Aligned Sentences
Do you have high blood pressure?
No diabetes? No high blood pressure?

Template Created
['do', 'you', 'have', '$0/JJ', '$1/NN', '$2/NN'], ['no', 'diabetes', '?', 'no', '$0/JJ', '$1/NN', '$2/NN']

Sentence Not Matched In Training Data
Do you have any ongoing medical problems?

Candidate Paraphrase Not Produced
*No diabetes? No any ongoing medical problems

Figure 9: An example of a poor candidate paraphrase which is no longer produced after implementing a POS restriction on single word variables.

Original Aligned Sentences

No diabetes or high blood pressure?
No diabetes? No high blood pressure?

Template Created

['$0/DT', '$1/NN', 'or', '$2/JJ', '$3/NN', '$4/NN'], ['$0/DT', '$1/NN', '?', '$0/DT', '$2/JJ', '$3/NN', '$4/NN']

Sentence Not Matched In Training Data

Any chronic illnesses or current medical problems?

Candidate Paraphrase Not Produced

Any chronic illnesses? Any current medical problems?

Figure 10: An example of a good candidate paraphrase which is no longer produced after implementing a POS restriction on single word variables.

sentences and matched sentences in the training data. Without POS tag restrictions the 'sentence not matched' in Figure 10 would in fact be matched and produce the good candidate paraphrase which follows because 'chronic illnesses' would be an acceptable match to variable '$1' even though they are not an acceptable match for variable '$1/NN'.

Restricting the length and POS tags for multiword variables reduced production of unique candidate paraphrases. Of the 1962 candidate paraphrases produced from these templates, only 257 are unique. The rate of uniqueness in candidate paraphrases is therefore 13 percent. Of these 257 candidate paraphrases, 36 are found in the Wilkins data and 41 percent of annotated examples are found to be good. Thus, for the 76 unique candidate paraphrases produced on rare labels, we would expect to see 31 true paraphrases. Figure 11 and Figure 12, show candidate paraphrases which are not produced only in this method. The candidate paraphrase in Figure 11 is not good and therefore not producing it increases precision again. However, the candidate paraphrase in

Original Aligned Sentences

Do you have any stds?
Ever have any sexually transmitted diseases?

Template Created

['$0/['VBP', 'VB']', 'you', '$1/DT', '$2/[NN]'], ['ever', '$0/[VBP]', '$1/DT', '$2/['RB', 'VBN', 'NNS']']

Sentence Not Matched In Training Data

Are you a mechanic?

Candidate Paraphrase Not Produced

*Ever are a mechanic?

Figure 11: An example of a poor candidate paraphrase which is no longer produced after implementing a POS restriction on all variables.

Original Aligned Sentences

Do you engage in physical activity?
What do you do for exercise?

Template Created

['$0/VB', 'you', '$1/['VB', 'IN']', '$2/['JJ', 'NN']'], ['what', '$0/VBP', 'you', '$1/['VB']', 'for', '$2/['NN']']

Sentence Not Matched In Training Data

Have you had surgeries in the past?

Candidate Paraphrase Not Produced

What have you had for surgeries in the past?

Figure 12: An example of a good candidate paraphrase which is no longer produced after implementing a POS restriction on all variables.

Figure 112 is good, so not producing it reduces recall again. In fact the 'IN' POS tags requirement in variable '$1' will usually produce odd paraphrases. For example, 'any' matches to this requirement in the similar sentence *Have you had any surgeries in the past* but this produces the paraphrase *What have you had for any surgeries in the past* which is odd grammar. In this way, this specific template may effect precision when restricted by POS tags.

Templates with POS tag restricted single word variables and multiword variables only restricted by length produced 3697 candidate paraphrases. Of these, 9 percent were unique meaning 335 unique candidate paraphrases were produced. The same 36 paraphrases from the last method are again found in the Wilkins data and 38 percent of annotated examples are found to be true paraphrases. We would expect this method to produce 44 good paraphrases on rare labels. Additionally, this method improves recall because the variables are less restrictive. An example of a good paraphrase which was lost in the method with POS restrictions on all variables but which is recovered in this method is given in Figure 13

Original Aligned Sentences

What helps?
What have you tried so far?

Template Created

['what.', '$0/[1]'], ['what', '$0/[3]', 'so', 'far']

Sentence Matched In Training Data

What else?

Candidate Paraphrases Produced

What else so far?

Figure 13: An example of a good candidate paraphrase which is no longer produced after implementing a POS restriction on all variables.

Original Aligned Sentences

Okay. How many days ago did this happen?
Ok, and when did this incident happen?

Template Created

['okay.', '$0', '$1/VBD', '$2', '$3/VB'], ['ok,', 'and', '$0', '$1/VBD', '$2', '$3/VB']

Sentence Matched In Training Data

Okay. How long ago did this happen?

Candidate Paraphrases Produced

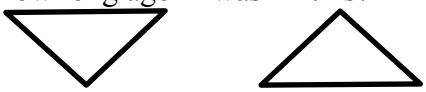
*Ok, and when's that was?
*Ok, and how long ago began you you?
*Ok, and when with your friend?

Figure 14: An example of a good candidate paraphrase which is no longer produced after implementing a POS restriction on all variables.

Combining templates with only POS tag restrictions on single word variables and variable swapping from gold alignments on multiword variables produced 30,059 unique candidate paraphrases. This number, added to the number of candidate paraphrases produced from templates where only single variables have POS tag restrictions, becomes extremely comparable to the number of candidate paraphrases produced by templates with no restrictions. Of the over 30,000 candidate paraphrases produced, 6 are found in the Wilkins 94 dialogues dataset. Annotating 387 of these paraphrases we find 8 to be acceptable. This means this method has 2% precision and we estimate it would produce about 600 novel, acceptable paraphrases. 7708 of these paraphrases were produced on rare labels, meaning with consistent precision we would expect to see 154 novel, good paraphrase on rare labels.

To asses why the precision of this method was so poor we look at an example in Figure 14. We see here that odd alignments from the data cause problems in the new setting of these paraphrases. For example, matching the sentence *Okay. How long ago did this happen?* causes 'this' to match the '$2' variable, and it is replaced with strings aligned elsewhere to 'this'. These aligned strings include 'that', 'you', and 'with your friend', all of which create poor paraphrases, and illustrate that the standard of similarity among aligned phrases is not as high as we hoped when implementing this method. However, these alignments are still good, for example 'this' aligned to 'with your friend' comes from the aligned sentences in N).

N) How long ago was this?



When were you with your friend?

7. CONCLUSIONS AND FUTURE WORK

There seems to be some value in implementing POS tag restriction on single word variables. Both in limiting possible matches, and altering words to fit the intend POS needed to make a proper paraphrase, we gain 10 percent accuracy over templates with no restrictions. We also find that restrictions on multiword variables produce gains of 3 to 6 percent. However, each restriction also significantly reduces the recall of potential, good paraphrases. We are optimistic that every method produces good, novel paraphrases on rare labels, but we see the immediate need for efforts to expand the recall of this approach.

It seems that when using only gold alignments, making restrictions on our question data hinders paraphrase recall more significantly than it aids precision. However, there is some value gained in performing variable swaps using an alignment dictionary. Precision is reduced significantly, but we are able to double the overall gain of potentially good paraphrases compared to the templates with POS tag restricted single word variables and no variable swapping when implementing this method.

Our aim for these paraphrases was to augment the rare label data. We want this new data to have high enough recall that rare labels have comparably-sized sets of training data as other examples. We also want the produced paraphrases to be precise enough that they are as useful in training the CNN stack as natural data would be. The balance of recall and precision is best represented in the number of expected good paraphrases on rare labels. Because the method with no variable restrictions has the highest value in this parameter, we would expect it to have the most positive effect on the downstream task of improving the virtual patient dialogue system. If we wanted to augment the rare label data to the point that each label had 13 examples, which is the overall average for labels, we would need to augment the rare label data with almost 2650 good

paraphrases. The only method which reaches close to this volume of production is the method with no restrictions on paraphrases.

The variable switching is an extremely noisy procedure. Currently, we take no account of how swapping a variable for something it is aligned to in the data effects the paraphrases until their evaluation. We see value in using ELMo representations as an intermediate step to reduce noise generated in the variable switching process. Because ELMo representations are contextualized, we can compare the same string in different sentences and find changes in its representation. We could compare the representation for a variable in the context of a question, and its representation in a paraphrase. If the representations are too dissimilar we would prevent the production of that paraphrase.

Overall, we would hope to see more useful data to augment the virtual patient dialogue system. We believe the next step in completing that process is to further pursue automatic alignments. We see that these alignments produce around 600 times as many templates as manual alignments do. Our expectation is that this method will also produce significant noise. However, we believe that the noise in our current data, as well as in future data, should be alleviated by a proper mechanism for picking good paraphrases out of all those produced. David King is in the process of developing a metric to do just that, which ranks the quality of paraphrases. We find that for our larger datasets with high recall, this would be a way to offset low precision.

In the future, we suggest exploring the use of dependency parsing in inducing multiword alignments using ELMo. We believe dependencies may suggest which words within a sentence could be part of a group to be aligned across sentences. We cite examples including split participle verbs and aligned subject noun phrases where one subject is a pronoun and the other is multiple words as evidence for our hypothesis. These suggestions are based on methods from human annotation standards (Thadani et al., 2012; Yao et al., 2013). Having multiword alignments included in our automatic alignments will improve our confidence in them because they will model the gold standard more closely.

We also suggest the further exploration of variable swapping and evaluation of variable swaps. This is another useful method for producing higher numbers of paraphrases. With an in-

crease in the accuracy of automatic alignments, the dictionaries used for this variable swapping will be concurrently improved.

ACKNOWLEDGEMENTS

This work was motivated and supported by the entire virtual patient dialogue system team. Special thanks to Amad Hussein for providing the automatic alignments and David King for his work in providing me with many important resources and running downstream evaluations for me, as well as for his advice on my work. My work would have been completely impossible if not for my advisor Michael White. A million thanks go to him for his work in mentoring me on this project and his guidance on writing this thesis. Thanks also to my thesis defense board members Micha Elsner, and Douglas Danforth. Additionally, part of this work was supported but the National Science Foundation's Research Experience for Undergraduate's Fellowship.

REFERENCES

Douglas Danforth, A. Price, K. Maicher, D. Post, B. Liston, D. Clinchot, C. Ledford, D. Way, and H. Cronau. 2013. Can virtual standardized patients be used to assess communication skills in medical students. In Proceedings of the 17th Annual IAMSE Meeting, St. Andrews, Scotland.

Douglas Danforth, Mike Procter, Richard Chen, Mary Johnson, and Robert Heller. 2009. Development of virtual patient simulations for medical education. Journal For Virtual Worlds Research 2(2).

Anthony Fader, Luke Zettlemoyer, Oren Etzioni. 2013. Paraphrase-Driven Learning for Open Question Answering. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1608-1618.

Anthony Fader, Luke Zettlemoyer, Oren Etzioni. 2014. Open Question Answering Over Curated and Extracted Knowledge Bases. Conference on Knowledge Discovery and Data Mining (KDD).

Ajda Gokcen, Evan Jaffe, Johnsey Erdmann, Michael White, Douglas Danforth. 2016. A Corpus of Word-Aligned Asked and Anticipated Questions in a Virtual Patient Dialogue System. Proceedings of the 10th Edition of the Language Resources and Evaluation Conference (LREC 2016).

Lifeng Jin, Michael White, Evan Jaffe, Laura Zimmerman, Douglas Danforth. 2017. Combining CNNs and Pattern Matching for Question Interpretation in a Virtual Patient Dialogue System. Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, page 11-21.

Lifeng Jin, David King, Amad Hussein, Michael White, Douglas Danforth. 2018. Using Paraphrasing and Memory-Augmented Models to Combat Data Sparsity in Question Interpretation with a Virtual Patient Dialogue System. Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 13-23.

Jonathon Mallison, Rico Sennrich, Mirella Lapata. 2017. Paraphrasing Revisited with Neural Machine Translation. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 881-893.

Franz Josef Och, Hermann Ney. "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In North American Association for Computational Linguistics (NAACL).

K. Thadani, S. Martin, and Michael White, (2012). A joint phrasal and dependency model for paraphrase alignment. In Proceedings of COLING 2012: Posters, pages 1229– 1238, Mumbai, India, December. The COLING 2012 Organizing Committee.

Wilcox, B. (2011). Chatscript. http://chatscript.sourceforge.net/

X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark, (2013). A lightweight and high performance monolin- gual word aligner. In Proceedings of ACL short.